REMARKS

Claim rejections under 35 USC 103(a)

Claims 1-10, 12-19, and 21-30 have been rejected under 35 USC 103(a) as being unpatentable over Kiick (2003/0200250) in view of Rowlands (2003/0200250). Claims 1, 7, 16, 21, and 25 are independent claims, from which the remaining claims ultimately depend. Applicant submits that the independent claims as amended are patentable, such that the remaining claims that are still pending and that have been rejected on this basis are patentable at least because they depend from independent base claims. Throughout the following discussion, claim 1 is discussed as representative of all the independent claims insofar as the rejection over Kiick in view of one or more other references in combination is concerned.

Based on the Examiner's statements, it appears that Rowlands is being relied upon just to show processorless nodes, and that otherwise the Examiner relies on Kiick as teaching all the limitations of the claimed invention. Applicant has amended the claimed invention so that some of the nodes are both "processorless and/or memoryless." Therefore, Applicant presumes that the Examiner would reject the claims over Kiick in view of Carpenter (cited in the previous office action) and/or Rowlands, insofar as Carpenter teaches processorless nodes, and Rowlands seems to teach memoryless nodes.

However, Applicant would like to direct the Examiner's attention to something else. Applicant has review the final office action in detail, and believes that the limitations of claims 2 and 3 are not actually suggested by Kiick, and thus not suggested by Kiick in view of Carpenter and/or Rowlands. That said, Applicant understands that the Examiner's interpretation of claims 2 and 3 may be a bit different than the way Applicant interprets them. Therefore, Applicant (1) has incorporated the limitations of claims 2 and 3 into claim 1, but also (2) has amended these limitations so that Applicant's interpretation is the only way you can interpret these claims – i.e., the claim limitations of claims 2 and 3 have been modified a bit when incorporated into claim 1 so that the Examiner and Applicant are "on the same page" as to interpretation of these claim limitations.

Therefore, Applicant now discusses what is going on in claim 1. First, what is specifically delineated is the process by which the interrupts for the I/O devices are assigned among the nodes of the system. In particular, this process is delineated "for a given interrupt for a given I/O device" to make it easier to understand the process. Now, "the given I/O device is connected to a first node of the nodes of the system," where such a "first node" is specifically delineated also to clarify the claim. Furthermore, "the given I/O device ha[s] an interrupt service routine to handle the given interrupt," where specifically "the interrupt route resid[es] at a second node of the nodes of the system." Again, such a "second node" is specifically delineated to clarify the claim. Also, and importantly, "*the first node [is] different than the second node*." That is, the first node is not the same as the second node, but is a different node. You can see this in FIG. 2 of the patent application, where the node 202A has the I/O device 206A connected thereto but the ISR 208A for this device resides at the node 202C, and the node 202A is different than the node 202C (i.e., they are not the same node).

Next, the process by which the given interrupt for the given I/O device is assigned to a node is as follows; it is noted that this process corresponds to the process described on page 6, lines 11-22 of the patent application as filed. First, "if the first node (to which the given I/O device is connected) has a cache, memory, and at least one processor," then the given interrupt is assigned to the first node. Therefore, in the example of FIG. 2, if the node 202A has a cache, memory, and at least one processor, then you assign the given interrupt to this node 202A for processing.

However, second, "if the first node *does not* have a cache, memory, and at least one processor, *but* the second node (at which the interrupt service routine for the given I/O device resides) *does* have a cache, memory, and at least one processor," then the given interrupt is assigned to the second node. Therefore, in the example of FIG. 2, if the node 202A does not have a cache, memory, and at least one processor, but the node 202C *does* have a cache, memory, and at least one processor, then you assign the given interrupt to this node 202C for processing.

Finally, third "if *both* the first node and the second node do not each have a cache, memory, and at least one processor," then you simply assign the given interrupt to a different (third) node that does have a cache, memory, and at least one processor, where the third node is not the same as either the first node or the second node. Therefore, in the example of FIG. 2, if the node 202A does not have a cache memory, and at least one processor, and the node 202C does not have a cache, memory, and at least one processor, then you assign the given interrupt to node 202B or node 202D, assuming one of which has a cache, memory, and at least one processor.

Now, in rejecting claims 1, 2, and 3 over Kiick in view of Rowlands (and presumably and/or in view of Carpenter as well), the Examiner has essentially relied upon paragraph [0034] of Kiick as teaching this assignment aspect of the claimed invention. However, note that the claimed invention is limited to three if/then clauses. If the first node (i.e., the node to which the I/O device is connected) has cache/memory/processors, then you assign the interrupt to this node. If the first node does not have cache/memory/processors, but the second node (i.e., the node at which the interrupt service routine resides) does have cache/memory/processors, then you assign the interrupt to this node. And if neither the first node nor the second node has cache/memory/processors, then you assign the interrupt to a completely different node that does have cache/memory/processors.

By comparison, paragraph [0034] of Kiick just says the following: "the interrupts need to be re-assigned to the 'closest' processors," period. Kiick, therefore, just does not teach this element of the invention. In the first instance, Kiick is talking about *re*-assignment here, not assignment; as such, paragraph [0034] is more relevant to the "dynamically modifying assignments" element of the claimed invention, not the element of the claimed invention that has been the subject of this discussion so far.

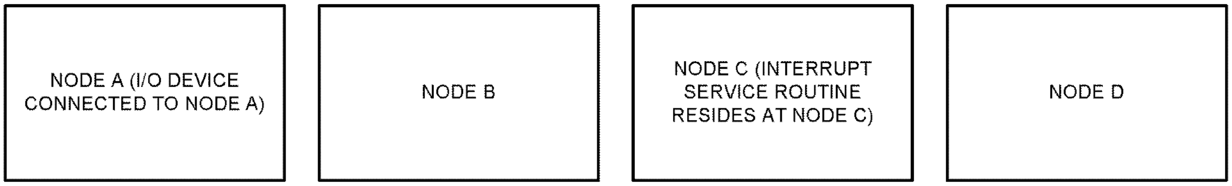
But in any case, assigning interrupts to the "closest" processor does not teach, disclose, or suggest three if/then clauses, as to which the claimed invention is now limited. The Examiner has interpreted paragraph [0034] as meaning that the interrupts are assigned to nodes to which the I/O devices are connected or to the nodes at which the ISR's assigned. (Office action, p. 3.) Applicant agrees only to a limited extent with the Examiner. That is, one of ordinary skill within the art may interpret the assignment of an interrupt to the closest processor in Kiick as meaning that the closest processor is defined physically, as the node that is closest to the interrupt, and thus the node that is connected to the I/O device. Indeed, Kiick in fact suggests as much, saying in paragraph [0034] that "[i]n other words, it is preferred that the interrupts are not assigned node or cell boundaries," which suggests a physical definition of closest.

However, the foregoing discussion is a bit academic, because Kiick does not teach, suggest, or disclose the actual limitations of the claimed invention. That is, the claimed invention says that "if the (first) node to which the I/O device is connected has cache/memory/processors, then assign the interrupt to this node; else if the (second) node at which the interrupt service routine resides has cache/memory/processors, then assign the interrupt to this node; else assign the interrupt to a completely different (third) node that does have cache/memory/processors." Kiick just says that you assign the interrupt to the closest processor. The closest processor, however, by definition is going to be the node to which the I/O device is connected – *always* – because the I/O device is connected to this node, and the interrupt comes from the I/O device.

Furthermore, in Kiick, all of the nodes are taught as having cache/memory/processors, so Kiick does not have to concern itself with the situation where this closest node does not have cache/memory/processors, in contradistinction to the claimed invention. Therefore, the next inquiry is if you modify Kiick so that one of the nodes does not have memory or processors (i.e., per Rowlands and/or Carpenter), how would Kiick operate? There is no teaching, suggestion, or disclosure within the prior art that one of ordinary skill within the art would be prompted to modify Kiick to particularly realize the claimed invention. For example, the first if/then clause of the claimed invention says that if the node to which the I/O device is connected has a cache, memory, *and* one or more processors, then you assign the interrupt to this node. If you do not use the claimed invention as a template to piece together and modify the prior art, however, Kiick could just as easily say that, well, so long as the node to which the I/O device is connected has a processor, then this node is still able to process the interrupt, and therefore we will still assign the interrupt to this node. The only reason that prompts one of ordinary skill within the art to assign the interrupt to the node to which the I/O device is connected *only if this node has all three of a cache, memory, and one or more processors* is found in the claimed invention, but that cannot be used as a reason to modify Kiick to yield the invention.

Furthermore, for sake of argument, let us even presume that Kiick per Rowlands and/or Carpenter can somehow be modified to teach this first if/then clause, where if the node to which the I/O device is connected has a cache, memory, and one or more processors, then you assign the interrupt to this node. Now, what if the node to which the I/O device is connected does not have a cache, memory, and one or more processors? Well, the claimed invention says that if this is case, but if the (second) node at which the interrupt service routine resides does have a cache, memory, and one or more processors, then you assign the interrupt to this (second) node. Again, however, if you do not use the claimed invention as a template to piece together and modify the prior art, Kiick could just as easily say that well, so long as the node at which the interrupt service routine resides has a processor, then this node is still able to process the interrupt, and therefore we will assign the interrupt to this node. The only reason that prompts one of ordinary skill within the art to assign the interrupt to the (second) node per the if/then clause that only if this node has all three of a cache, memory, and one or more processors is found in the claimed invention, but again that cannot be used as a reason to modify Kiick to yield the claimed invention.

Indeed, Applicant believes that the Examiner's interpretation of what a "closest" node is – as encompassing a node at which the interrupt service routine resides – does not make any sense in light of the ordinary meaning of "closeness" that one of ordinary skill within the art would ascribe to Kiick. For example, consider these four nodes:


NODE A (I/O DEVICE CONNECTED TO NODE A)
NODE B
NODE C (INTERRUPT SERVICE ROUTINE RESIDES AT NODE C)
NODE D

Now in the claimed invention, if node A has cache/memory/processors, then the interrupt is assigned to node A, because node A is connected to the I/O device generating the interrupt. But, if node A does not have cache/memory/processors, but if node C has cache/memory/processors, then the interrupt is assigned to node C, because the interrupt service routine resides at node C.

However, Kiick just says that you assign an interrupt to the "closest" node. This is node A, obviously, since node A is closest to the I/O device generating the interrupt, because node A is in fact connected to this I/O device. Kiick, nor any other reference of record does not say that you assign the interrupt node A only if node A has cache/memory/processors, and thus you Kiick in view of any other reference(s) does not teach this first if/then clause of the invention.

Furthermore, it is difficult to see how the "closest" node to the I/O device being generated in Kiick could even possibly be node C. And yet, if we are required to interpret Kiick such that the node at which the interrupt service routines resides receives assignment of the interrupt, as suggested by the Examiner, this is how one of ordinary skill within the art – without any benefit of using the claimed invention as a template to piece together or modify the prior art – would have to interpret Kiick. But this just does not make any sense.

For instance, the Examiner has not provided any reference or argument indicating how and/or why one of ordinary skill within the art would interpret "closest" as in Kiick to mean "not the closest" as is required by his interpretation. That is, if node C is somewhere "closer" to node A and the I/O device than node B is, Applicant respectfully submits that the Examiner could interpret any term cited in any reference to mean whatever he wants it to mean. A "cat" could be a "dog" (they're both mammals, aren't they?). Applicant does not mean to be facetious here, but where Kiick is very specific as *only* disclosing that an interrupt is assigned "to the 'closest' processor" it is difficult to see how these few words can somehow read on all of the following:

> wherein assigning the interrupts for the I/O devices among the nodes of the system comprises, for a given interrupt for a given I/O device, the given I/O device connected to a first node of the nodes of the system, the given I/O device having an interrupt service routine to handle the given interrupt, the interrupt service routine residing at a second node of the nodes of the system, the first node being different than the second node,
>
> if the first node to which the given I/O device is connected has a cache, memory, and at least one processor, then assigning the given interrupt for the given I/O device to the first node;
>
> if the first node does not have a cache, memory, and at least one processor, but the second node at which the interrupt service routine of the given I/O device resides does have a cache, memory, and at least one processor, then assigning the given interrupt for the given I/O device to the second node;
>
> if both the first node and the second node do not each have a cache, memory, and at least one processor, then assigning the given interrupt for the given I/O device to a third node of the nodes of the system, the third node having memory and at least one processor, the third node being different than the first node and the second node;

And yet, this is what is required for Kiick in view of any other reference(s) to suggest the claimed invention. There are no if/then clauses suggest in Kiick in view of any other reference(s), yet the claimed invention has three. Kiick just assigns an interrupt to the closest node, period.

Further, you cannot define "closest" as in Kiick to mean *both* physical proximity (such that node A in the example above is closest to the I/O device because it is connected to the I/O device) and the node at which the interrupt service routine resides (such that node C in the example above is closest to the I/O device because the interrupt service routine resides at this node). This is because you would then not be able to determine which is the closest node (singular) to which to assign an interrupt. For example, consider the following. I have an interrupt. Per Kiick, I am going to assign it to the closest node. I have nodes A and C per the example above. Which is the closest node – i.e., which node should I assign it to? The Examiner's interpretation of Kiick does help me, because it says that the closest node can mean either node A (as the node to which the I/O device is connected) *or* node C (as the node at which the interrupt service routine resides).

Thus, if you interpret Kiick as suggested by the Examiner, then Kiick in view of any other reference(s) basically teaches pick node A *or* node C – it does not really matter. However, in the claimed invention, it does matter! Specifically, if node A has memory, cache, and processors, then the interrupt is assigned to node A, period. Only if node A does not have memory, cache, and processors do you like to see if node C has memory, cache, and processors (and if node C does in this instance, then the interrupt is assigned to node C). For example, assume *both* nodes A and C each have memory, cache, and processors. Per the Examiner's interpretation, you could pick *either* node A or node C to which to assign the interrupt – it does not matter which node you pick, because either node satisfies the Examiner's interpretation that it is the node "closest" to the I/O device. However, in the claimed invention in this example, you *have* to pick node A, as specified by the first if/then clause.

All of the foregoing has not even touched upon the third if/then clause of the claimed invention. The third if/then clause says that if both the first and the second nodes (i.e., the node to which the I/O device is connected and the node at which the interrupt service routines reside) do not have cache/memory/processors, then you assign the interrupt to a completely different node that does have cache/memory/processors. To this end, saying that you assign an interrupt to the "closest" node, per Kiick, cannot mean that you assign the interrupt to a node that is <u>not</u> the closest node *by definition per the Examiner's interpretation of "closeness."* That is, the Examiner has said that the closest node is the first or the second node of the claimed invention. And yet, the claimed invention says that if the first and the second nodes each do not have cache/memory/processors, then you assign the interrupt to a third, completely different, node. However, since only the first and the second nodes are "closest" per Kiick, then you have to assign the interrupt to one of these nodes – and definitely not to a third node that is by definition <u>not</u> one of the closest nodes in Kiick. In other words, it is difficult to see how one of ordinary skill within the art would modify Kiick so that the resulting combination is *exactly the opposite* of what Kiick explicitly teaches – rather than assigning an interrupt to the closest node, you assign the interrupt to a node that is **not** the closest node.

For all of these reasons, therefore, Applicant submits that the claimed invention is patentable over Kiick in view of Rowlands and/or Carpenter.

Conclusion

Applicants have made a diligent effort to place the pending claims in condition for allowance, and request that they so be allowed. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Mike Dryja, Applicant's representative, at 425-427-5094, so that such issues may be resolved as expeditiously as possible. For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

September 20, 2007
Date

Michael A. Dryja, Reg. No. 39,662
Attorney/Agent for Applicant(s)

Law Offices of Michael Dryja
1474 N Cooper Rd #105-248
Gilbert, AZ 85233
tel: 425-427-5094
fax: 425-563-2098